



US009536340B2

(12) **United States Patent**  
**Loberg**

(10) **Patent No.:** **US 9,536,340 B2**  
(45) **Date of Patent:** **Jan. 3, 2017**

(54) **SOFTWARE INCORPORATING EFFICIENT 3-D RENDERING**

(71) Applicant: **DIRTT ENVIRONMENTAL SOLUTIONS, LTD.**, Calgary (CA)

(72) Inventor: **Barrie Loberg**, Millarville (CA)

(73) Assignee: **DIRTT Environmental Solutions, LTD.**, Calgary (CA)

(\*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 552 days.

(21) Appl. No.: **14/032,946**

(22) Filed: **Sep. 20, 2013**

(65) **Prior Publication Data**  
US 2014/0022243 A1 Jan. 23, 2014

**Related U.S. Application Data**

(63) Continuation of application No. 11/204,421, filed on Aug. 16, 2005, now Pat. No. 8,751,950.  
(Continued)

(51) **Int. Cl.**  
**G06T 15/00** (2011.01)  
**G06F 9/44** (2006.01)  
(Continued)

(52) **U.S. Cl.**  
CPC ..... **G06T 15/00** (2013.01); **G06F 9/4443** (2013.01); **G06F 17/50** (2013.01);  
(Continued)

(58) **Field of Classification Search**  
None  
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,510,369 A 4/1985 Harrison  
4,586,145 A 4/1986 Bracewell  
(Continued)

FOREIGN PATENT DOCUMENTS

EP 1098244 9/2001  
EP 1204046 5/2002  
(Continued)

OTHER PUBLICATIONS

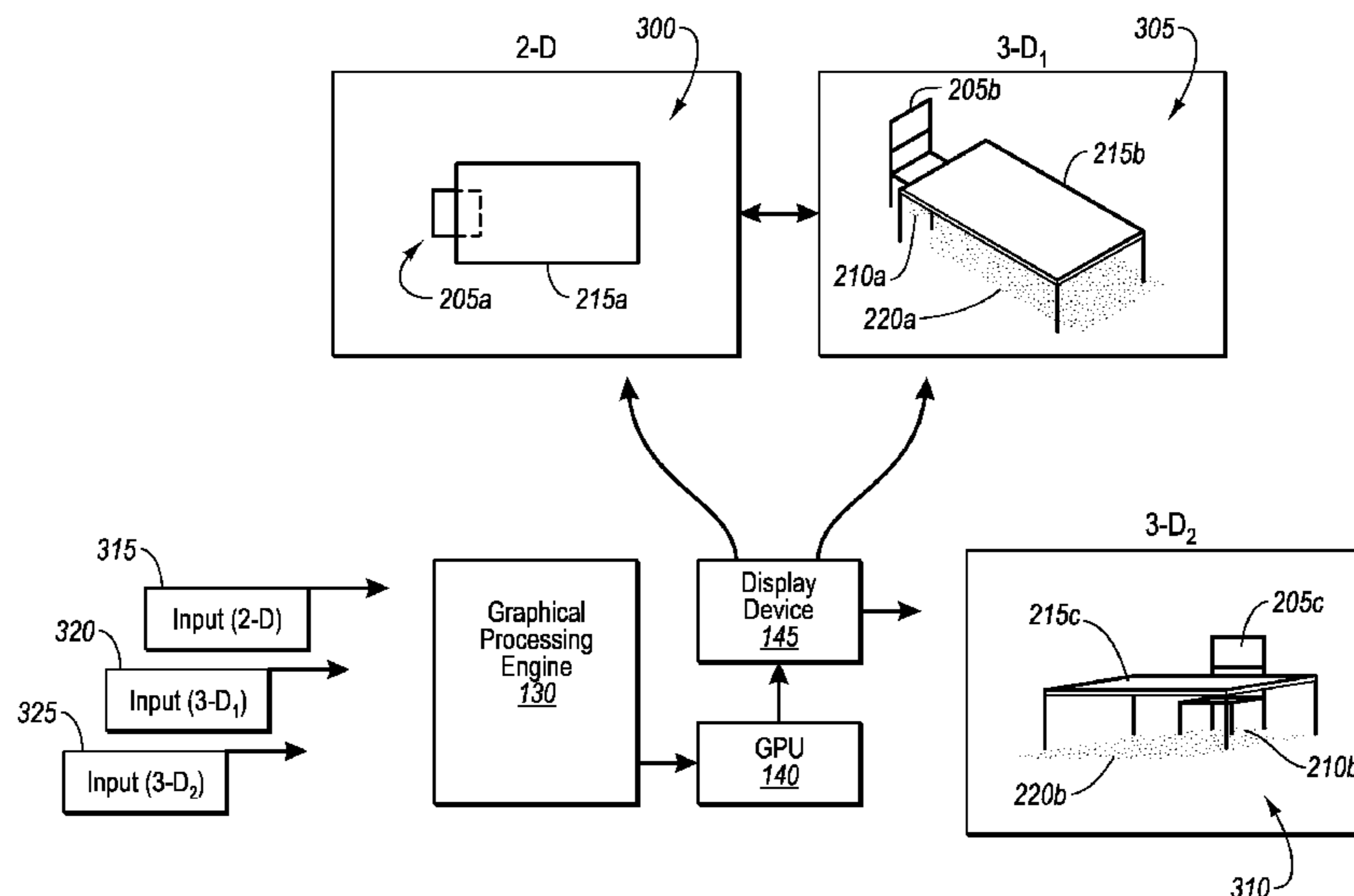
International Search Report and Witten Opinion for PCT/US2012/038582 mailed Dec. 28, 2012.  
(Continued)

*Primary Examiner* — Jeffrey A Gaffin  
*Assistant Examiner* — John M Heffington  
(74) *Attorney, Agent, or Firm* — Workman Nydegger

(57) **ABSTRACT**

Design software in accordance with an implementation of the present invention is configured to provide believable three-dimensional representations of user selections in real-time. Design elements that would otherwise be difficult to efficiently render three-dimensionally in real-time are pre-rendered for realistic visual effects, such as realistic shading, which correspond to various positions of the elements in a design space. Blanks of the visual effects for each position are then stored in a data store for visual effects. At run time, data associated with user design choices, as well as the blanks for any corresponding design elements are fed in one implementation to peripheral processing hardware, such as a GPU, which sends the processed data to a display device. The user is therefore able to view complex visual data of certain design choices efficiently with added realism.

**20 Claims, 6 Drawing Sheets**



<b>Related U.S. Application Data</b>					
		6,772,168	B2	8/2004	Ardoin
		6,813,610	B1	11/2004	Bienias
(60)	Provisional application No. 60/602,233, filed on Aug. 17, 2004.	6,826,539	B2 *	11/2004	Loveland ..... G06F 17/30286 705/4
		6,829,584	B2 *	12/2004	Loveland ..... G06Q 10/063 705/4
(51)	<b>Int. Cl.</b>	6,888,542	B1	5/2005	Clauss
	<i>G06F 17/50</i> (2006.01)	6,919,891	B2 *	7/2005	Schneider ..... G06T 17/005 345/419
	<i>G06T 19/20</i> (2011.01)				
(52)	<b>U.S. Cl.</b>	6,922,701	B1	7/2005	Ananian
	CPC ..... <i>G06F 17/5004</i> (2013.01); <i>G06F 2217/04</i> (2013.01); <i>G06T 19/20</i> (2013.01); <i>G06T 2200/24</i> (2013.01); <i>G06T 2210/04</i> (2013.01); <i>G06T 2210/61</i> (2013.01); <i>G06T 2219/2016</i> (2013.01)	6,944,513	B1	9/2005	Tomomitsu
		6,971,063	B1	11/2005	Rappaport
		6,985,832	B2	1/2006	Saebi
		6,999,102	B2	2/2006	Felser
		7,016,747	B1	3/2006	Ninomiya
		7,019,753	B2	3/2006	Rappaport
		7,042,440	B2	5/2006	Pryor
(56)	<b>References Cited</b>	7,050,955	B1 *	5/2006	Carmel ..... G06T 19/00 345/418
	<b>U.S. PATENT DOCUMENTS</b>	7,062,454	B1	6/2006	Giannini
		7,062,532	B1	6/2006	Sweat
		7,062,722	B1 *	6/2006	Carlin ..... G06T 15/10 715/781
	4,700,317 A 10/1987 Watanabe	7,065,420	B1	6/2006	Philpott
	4,829,446 A 5/1989 Draney	7,079,990	B2	7/2006	Haller
	4,862,376 A 8/1989 Ferriter	7,080,096	B1 *	7/2006	Imamura ..... G06Q 30/06
	5,111,392 A 5/1992 Malin	7,085,697	B1	8/2006	Rappaport
	5,255,207 A 10/1993 Cornwell	7,088,374	B2 *	8/2006	David ..... G06F 9/4443 345/440
	5,293,479 A 3/1994 Quintero	7,096,173	B1	8/2006	Rappaport
	5,339,247 A 8/1994 Kirihara	7,127,378	B2 *	10/2006	Hoffman ..... G01C 15/002 702/159
	5,386,507 A 1/1995 Teig	7,139,686	B1	11/2006	Critz
	5,414,801 A 5/1995 Smith	7,155,228	B2	12/2006	Rappaport
	5,514,232 A 5/1996 Burns	7,170,511	B2 *	1/2007	Sowizral ..... G06T 15/005 345/420
	5,555,357 A 9/1996 Fernandes	7,171,208	B2	1/2007	Rappaport
	5,555,366 A 9/1996 Teig	7,171,344	B2	1/2007	Lind
	5,572,639 A 11/1996 Gantt	7,173,623	B2 *	2/2007	Calkins ..... G06T 13/00 345/473
	5,576,965 A 11/1996 Akasaka	7,200,639	B1	4/2007	Yoshida
	5,588,098 A * 12/1996 Chen ..... G06F 3/04845 345/653	7,216,092	B1	5/2007	Weber
		7,218,979	B2	5/2007	Tsuji
	5,625,827 A 4/1997 Krause	7,243,054	B2	7/2007	Rappaport
	5,684,713 A 11/1997 Asada	7,246,044	B2 *	7/2007	Imamura ..... G06Q 30/0601 345/588
	5,740,341 A 4/1998 Oota	7,246,045	B1	7/2007	Rappaport
	5,742,294 A 4/1998 Watanabe	7,249,005	B2	7/2007	Loberg
	5,764,241 A 6/1998 Elliot	7,250,944	B2 *	7/2007	Anderson ..... G06F 11/328 345/419
	5,764,518 A 6/1998 Collins	7,262,775	B2 *	8/2007	Calkins ..... G06T 13/00 345/473
	5,849,440 A 12/1998 Lucas	7,266,768	B2	9/2007	Ferlitsch
	5,870,771 A 2/1999 Oberg	7,277,572	B2 *	10/2007	MacInnes ..... G06F 17/5004 345/419
	5,894,310 A 4/1999 Arsenault	7,277,830	B2	10/2007	Loberg
	5,918,232 A 6/1999 Pouschine	7,299,168	B2	11/2007	Rappaport
	5,977,982 A 11/1999 Lauzon	7,299,416	B2	11/2007	Jaeger
	5,995,107 A 11/1999 Berteig	7,318,063	B2	1/2008	Brychell
	6,014,503 A 1/2000 Nagata	7,337,093	B2	2/2008	Ramani
	6,020,885 A 2/2000 Honda	7,340,383	B2	3/2008	Mayuzumi
	6,025,847 A * 2/2000 Marks ..... G06T 19/20 345/419	7,353,192	B1	4/2008	Ellis
		7,392,522	B2	6/2008	Murray
	6,037,945 A 3/2000 Loveland	7,398,481	B2 *	7/2008	Kraus ..... G01C 15/00 345/419
	6,052,669 A 4/2000 Smith	7,430,711	B2	9/2008	Rivers-Moore
	6,253,167 B1 6/2001 Matsuda	7,437,376	B2	10/2008	Sikchi
	6,292,810 B1 9/2001 Richards	7,444,195	B2	10/2008	Smith
	6,295,513 B1 9/2001 Thackston	7,454,259	B2	11/2008	Ninomiya
	6,335,732 B1 1/2002 Shaikh	7,479,959	B2	1/2009	Han
	6,401,237 B1 6/2002 Ishikawa	7,492,934	B2	2/2009	Mundy
	6,459,435 B1 10/2002 Eichel	7,516,399	B2	4/2009	Hsu
	6,466,239 B2 * 10/2002 Ishikawa ..... G06T 15/10 715/850	7,523,411	B2 *	4/2009	Carlin ..... G06Q 30/02 345/419
		7,574,323	B2	8/2009	Rappaport
	6,483,508 B1 11/2002 Ishikawa	7,587,302	B2	9/2009	Arvin
	6,493,679 B1 12/2002 Rappaport	7,596,518	B2	9/2009	Rappaport
	6,509,906 B1 1/2003 Awe	7,620,638	B2	11/2009	Nonclercq
	6,552,721 B1 * 4/2003 Ishikawa ..... G06F 3/14 345/418				
	6,570,563 B1 5/2003 Honda				
	6,626,954 B1 * 9/2003 Kamachi ..... G06T 13/205 345/473				
	6,662,144 B1 12/2003 Normann				
	6,684,255 B1 1/2004 Martin				
	6,690,981 B1 2/2004 Kawachi				
	6,701,288 B1 3/2004 Normann				
	6,721,684 B1 4/2004 Saebi				
	6,734,852 B2 * 5/2004 Sowizral ..... G06T 15/005 345/420				

**US 9,536,340 B2**

(56)

**References Cited**

U.S. PATENT DOCUMENTS

			2001/0047251	A1	11/2001	Kemp	
			2002/0010589	A1	1/2002	Nashida	
			2002/0065635	A1*	5/2002	Lei .....	G06T 19/00 703/1
7,629,985	B2	12/2009	McArdle		6/2002	Rao	
7,643,027	B2	1/2010	Rothstein		6/2002	Wucherer	
7,643,966	B2	1/2010	Adachi		7/2002	Ishikawa	
7,661,959	B2	2/2010	Green		7/2002	Ferlitsch	
7,676,348	B2	3/2010	Okada		7/2002	Carlin .....	G06Q 30/02 715/778
7,761,266	B2	7/2010	Mangon				
7,788,068	B2	8/2010	Mangon		2002/0095348	A1	7/2002 Hiroshige
7,814,436	B2	10/2010	Schrag		2002/0116163	A1*	8/2002 Loveland .....
7,822,584	B1	10/2010	Saebi				G06F 17/30286 703/13
7,823,074	B2	10/2010	Takemura		2002/0144204	A1	10/2002 Milner
7,856,342	B1	12/2010	Kfourri		2002/0158865	A1	10/2002 Dye
7,864,173	B2*	1/2011	Handley .....	G06F 3/016 345/420	2002/0188678	A1	12/2002 Edecker
					2002/0196285	A1	12/2002 Sojoodi
7,877,237	B1	1/2011	Saebi		2003/0097273	A1	5/2003 Carpenter
7,996,756	B2	8/2011	Eilers		2004/0012542	A1	1/2004 Bowsher
8,065,623	B2	11/2011	Bohlman		2004/0027371	A1	2/2004 Jaeger
8,065,654	B2	11/2011	Shiihara		2004/0059436	A1*	3/2004 Anderson .....
8,100,552	B2*	1/2012	Spero .....	B60Q 1/04 362/227			G06F 11/328 700/2
					2004/0098691	A1	5/2004 Teig
8,108,267	B2	1/2012	Varon		2004/0104934	A1*	6/2004 Fager .....
8,117,558	B2	2/2012	Hoguet				A63F 13/10 715/757
8,132,123	B2	3/2012	Schrag		2004/0117746	A1	6/2004 Narain
8,134,553	B2	3/2012	Saini		2004/0145614	A1	7/2004 Takagaki
8,185,219	B2	5/2012	Gilbert		2004/0153824	A1	8/2004 Devarajan
8,195,434	B2	6/2012	Powell		2004/0204903	A1	10/2004 Saebi
8,244,025	B2	8/2012	Davis		2004/0205519	A1	10/2004 Chapel
8,255,338	B1	8/2012	Brittan		2004/0236561	A1	11/2004 Smith
8,270,769	B2	9/2012	Judelson		2005/0041028	A1	2/2005 Coutts
8,271,336	B2	9/2012	Mikurak		2005/0065951	A1	3/2005 Liston
8,276,088	B2	9/2012	Ke		2005/0071135	A1	3/2005 Vredenburgh
8,285,707	B2	10/2012	Day		2005/0081161	A1*	4/2005 MacInnes .....
8,290,849	B2	10/2012	Eisler				G06F 17/5004 715/765
8,301,527	B2	10/2012	Tarbox		2005/0140668	A1*	6/2005 Hlavac .....
8,314,799	B2	11/2012	Pelletier				G06T 13/00 345/419
8,326,926	B2	12/2012	Sangem		2005/0203718	A1	9/2005 Carek
8,332,401	B2	12/2012	Hull		2005/0256874	A1	11/2005 Chiba
8,332,827	B2	12/2012	Edde		2006/0028695	A1	2/2006 Knighton
8,334,867	B1	12/2012	Davidson		2006/0041518	A1	2/2006 Blair
8,335,789	B2	12/2012	Hull		2006/0041842	A1	2/2006 Loberg
8,352,218	B2	1/2013	Balla		2006/0119601	A1	6/2006 Finlayson
8,386,918	B2	2/2013	Do		2006/0143220	A1	6/2006 Spencer
RE44,054	E	3/2013	Kim		2006/0174209	A1	8/2006 Barros
8,402,473	B1	3/2013	Becker		2006/0206623	A1	9/2006 Gipps
8,423,391	B2	4/2013	Hessedenz		2006/0271378	A1	11/2006 Day
8,442,850	B2	5/2013	Schorr		2007/0065002	A1	3/2007 Marzell
8,462,147	B2	6/2013	Sugden		2007/0088704	A1	4/2007 Bourne
8,468,175	B2	6/2013	Obata		2007/0097121	A1	5/2007 Loop
8,499,250	B2	7/2013	Wetzer		2007/0115275	A1	5/2007 Cook
8,508,539	B2	8/2013	Vlietinck		2007/0168325	A1	7/2007 Bourne
8,510,382	B2	8/2013	Purdy		2007/0180425	A1	8/2007 Storms
8,510,672	B2	8/2013	Loberg		2007/0188488	A1	8/2007 Choi
8,521,737	B2	8/2013	Hart		2007/0204241	A1	8/2007 Glennie
8,533,596	B2	9/2013	Boss		2007/0219645	A1	9/2007 Thomas
8,566,419	B2	10/2013	Purdy		2007/0240049	A1	10/2007 Rogerson
8,572,558	B1	10/2013	Mathieu		2007/0250295	A1	10/2007 Murray
8,583,375	B2	11/2013	Baule		2007/0260432	A1	11/2007 Okada
8,600,989	B2	12/2013	Hull		2007/0271870	A1	11/2007 Mifsud
8,626,877	B2	1/2014	Greene		2007/0294622	A1	12/2007 Sterner
8,645,973	B2	2/2014	Bosworth		2008/0036769	A1	2/2008 Coutts
8,650,179	B2	2/2014	Driesch		2008/0052618	A1	2/2008 McMillan
8,763,009	B2	6/2014	Degirmenci		2008/0126021	A1	5/2008 Hoguet
8,773,426	B1	7/2014	Hamel		2008/0140732	A1	6/2008 Wilson
8,819,072	B1	8/2014	Croicu		2008/0141334	A1	6/2008 Wicker
8,914,259	B2	12/2014	Kripac		2008/0143884	A1	6/2008 Foster
8,933,925	B2	1/2015	Sinha		2008/0165183	A1	7/2008 Rassieur
8,949,789	B2	2/2015	Schlarb		2008/0174598	A1	7/2008 Risenhoover
8,954,295	B1	2/2015	Vicknair		2008/0188969	A1	8/2008 O'Malley
8,994,726	B1	3/2015	Furukawa		2008/0238946	A1	10/2008 Baughman
9,075,931	B2	7/2015	Charles		2008/0249756	A1	10/2008 Chaisuparasmikul
9,106,425	B2	8/2015	Murphey		2008/0275674	A1	11/2008 Reghetti
9,117,308	B1	8/2015	Nag		2008/0282166	A1	11/2008 Fillman
2001/0024211	A1	9/2001	Kudukoli		2008/0303844	A1	12/2008 Reghetti
2001/0024230	A1	9/2001	Tsukahara		2008/0309678	A1	12/2008 Reghetti
2001/0047250	A1*	11/2001	Schuller .....	G06F 17/5004 703/1	2009/0069704	A1	3/2009 MacAdam
					2009/0110307	A1	4/2009 Markowitz

(56)

## References Cited

## U.S. PATENT DOCUMENTS

2009/0113349	A1	4/2009	Zohar
2009/0119039	A1	5/2009	Banister
2009/0138826	A1	5/2009	Barros
2009/0148050	A1	6/2009	Reggetti
2009/0187385	A1	7/2009	Zegdoun
2009/0210487	A1	8/2009	Westerhoff
2009/0248184	A1	10/2009	Steingart
2009/0254843	A1	10/2009	Van Wie
2009/0273598	A1	11/2009	Reggetti
2010/0017733	A1	1/2010	Barros
2010/0042516	A1	2/2010	Knipfer
2010/0121614	A1	5/2010	Reggetti
2010/0122196	A1	5/2010	Wetzer
2010/0138762	A1	6/2010	Reggetti
2010/0185514	A1	7/2010	Glazer
2010/0302245	A1	12/2010	Best
2011/0078169	A1	3/2011	Sit
2011/0169826	A1	7/2011	Elsberg
2011/0258573	A1	10/2011	Wetzer
2011/0320966	A1	12/2011	Edecker
2012/0005353	A1	1/2012	Edecker
2012/0069011	A1	3/2012	Hurt
2012/0331422	A1	12/2012	High
2014/0067333	A1	3/2014	Rodney

## FOREIGN PATENT DOCUMENTS

FR	2563516	10/1985
GB	120633	11/1918
GB	2364801	2/2002
JP	62127426	6/1987
JP	10334127	12/1998
JP	2005301630	10/2005
WO	9003618	4/1990
WO	9322741	11/1993
WO	0219177	3/2002
WO	02075597	9/2002
WO	03/023559	3/2003
WO	2005033985	4/2005
WO	2006018744	2/2006
WO	2007093060	8/2007
WO	2007106873	9/2007
WO	2006018740	2/2009
WO	2009100538	8/2009

## OTHER PUBLICATIONS

Gajamani, Geetha., "Automated Project Scheduling and Inventory Monitoring Using RFID", 2007, 24th International Symposium on Automation and Robotics in Construction. Retrieved from [http://www.iaarc.org/publications/proceedings\\_of\\_the\\_24th\\_isarc/automated\\_project\\_schedule\\_and\\_inventory\\_monitoring\\_using\\_rfid.html](http://www.iaarc.org/publications/proceedings_of_the_24th_isarc/automated_project_schedule_and_inventory_monitoring_using_rfid.html).

European Search Report—Application No. 10 833 972.2 mailed Feb. 2, 2015.

European Search Report—Application No. 12800327.4 mailed Feb. 18, 2015.

Funkhouser T et al: Modeling by example, ACM Transactions on Graphic (TOG), ACM, US, vol. 23, No. 3, Aug. 1, 2004 (Aug. 1,

2004), pp. 652-663, XP008087089, ISSN: 0730-0301, DOI: 10.1145/1015706.1015775 \*abstract\* \*p. 652, col. 1, paragraph 2—col. 2, paragraph 2\* \*p. 656, col. 1, paragraph 3-5\*.

Anonymous: "Dresden Frauenkirche—Wikipedia, the free encyclopedia", Jun. 8, 2011 (Jun. 8, 2011), pp. 1-7, XP055168540 retrieved from the Internet: URL: [https://en.wikipedia.org/w/index.php?title=Dresden\\_Frauenkirche&oldid=433207596](https://en.wikipedia.org/w/index.php?title=Dresden_Frauenkirche&oldid=433207596) [retrieved on Feb. 9, 2015] \*p. 3, paragraph 7-9\*.

Wollongong, City Centre 3D Model Specifications, May 18, 2009. Notice of Allowance for U.S. Appl. No. 13/510,912 mailed Sep. 14, 2015.

Chan, et al.: "Design of a Walkthrough System for Indoor Environments from Floor Plans"; Proceedings of the 1998 IEEE Conference on Information Visualization, Jul. 29-31, 1998, pp. 50-57. Edward J. Dejesus, James P. Callan, and Curtis R. Whitehead, Pearl: An Expert System for Power Supply Layout, 23rd Design Automation Conference, Paper 34.4, IEEE.

Josie Wernecke; Title: The Inventor Mentor: Programming Object Oriented 3D Graphics with Open Inventor; Release 2; Date: Jun. 19, 1997; Published on Web Site: [www.cs.ualberta.ca/](http://www.cs.ualberta.ca/).

International Search Report and Opinion on PCT/CA2007/000241, mailed May 15, 2007.

International Search Report and Opinion on PCT/CA2009/000190, mailed Jun. 5, 2009.

International Search Report and Opinion on PCT/CA2009/000183, mailed Jun. 9, 2009.

International Search Report and Opinion on PCT/CA2009/000311, mailed Jul. 30, 2009.

International Search Report & Written Opinion for PCT/US2010/058092 dated Jul. 27, 2011.

Marir F et al: "OSCONCAD: a model-based CAD system integrated with computer applications", Proceedings of the International Construction IT Conference, vol. 3, 1998.

EPO Search Report for EP07701791 dated Aug. 21, 2012.

International Search Report for PCT/CA2009000190 mailed Apr. 12, 2012.

Blythe, Rise of the Graphics Processor, IEEE, 2008.

U.S. Office of Personnel Management, Clear Cache, IE, 2007.

Wang, Intellectual Property Protection in Collaborative Design through Lean information Modeling and Sharing, 2006.

Lea, Community Place Architecture and Performance, 2007.

Autodesk, Maya 8.5 Shading, 2007.

CRC, Final Report Collaboration Platform. 2009.

Mental Images GmbH, RealityServer Functional Overview White Paper, 2007.

Rozansk, Software Systems Architecture Working With Stakeholders Using Viewpoints and Perspectives, 2008.

Sony, Community Place Conductor 2.0 Users Manual, 1998.

Vasko, Collaborative Modeling of Web Applications for Various Stakeholders, 2009.

Gross M D: "Why can't CAD be more like Lego? CKB, a program for building constructions kits", Automation in Construction, Elsevier Science Publishers, Amsterdam, NL, vol. 5, No. 4, Oct. 1, 1996 (Oct. 1, 1996) pp. 285-300, XP004072247, ISSN: 0926-5805, DOI:10.1016/S0926-5805(96)00154-9.

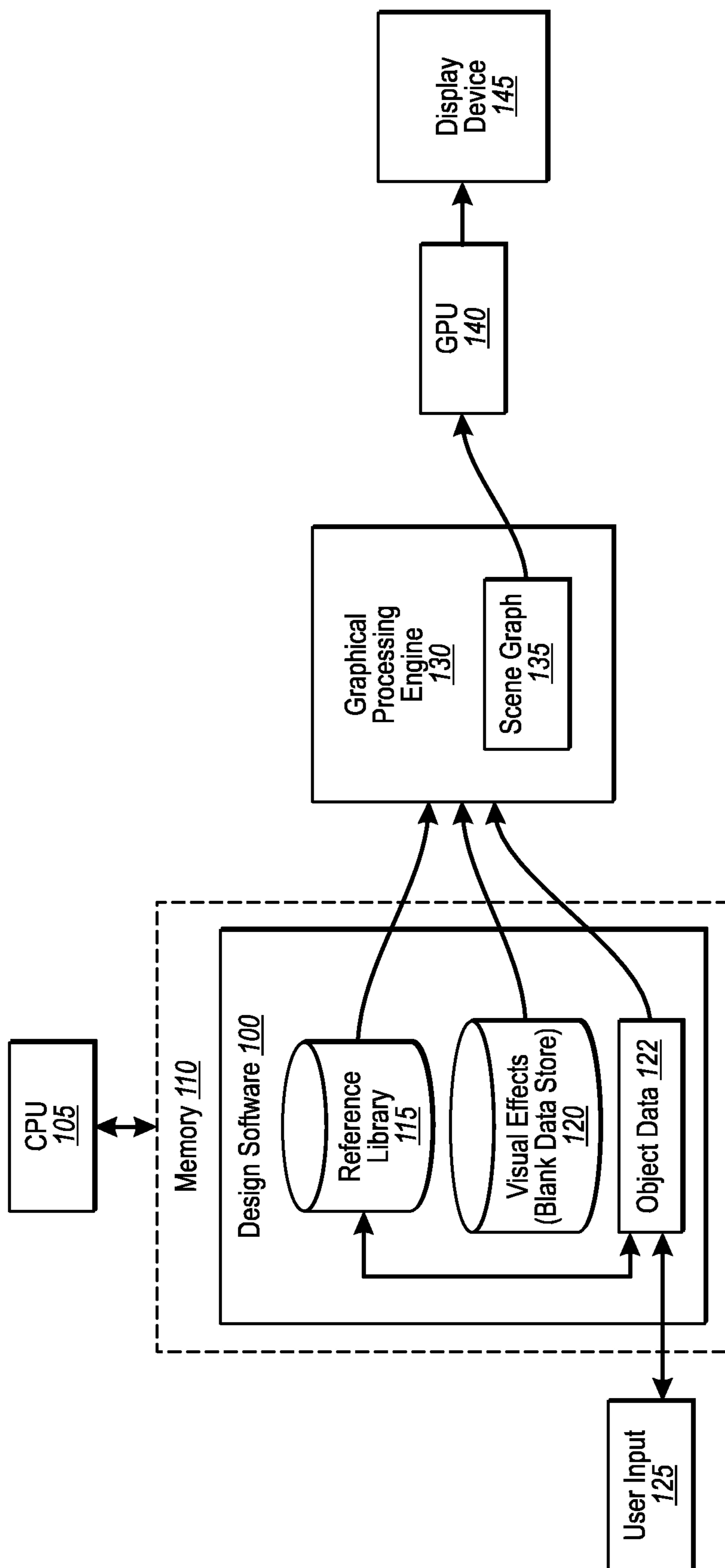
European Search Report—Application No./ Patent No. 09719352. 8-1960 / 2252951 PCT/CA2009000311.

Notice of Allowance for U.S. Appl. No. 13/579,261 mailed on Jul. 10, 2015.

Bing search q=cad+component+replicate&src=IE-Sea Apr. 2, 2016.

Bing search q=cad+replicate+part&src=IE-SearchBo Apr. 2, 2016.

\* cited by examiner



**Fig. 1**

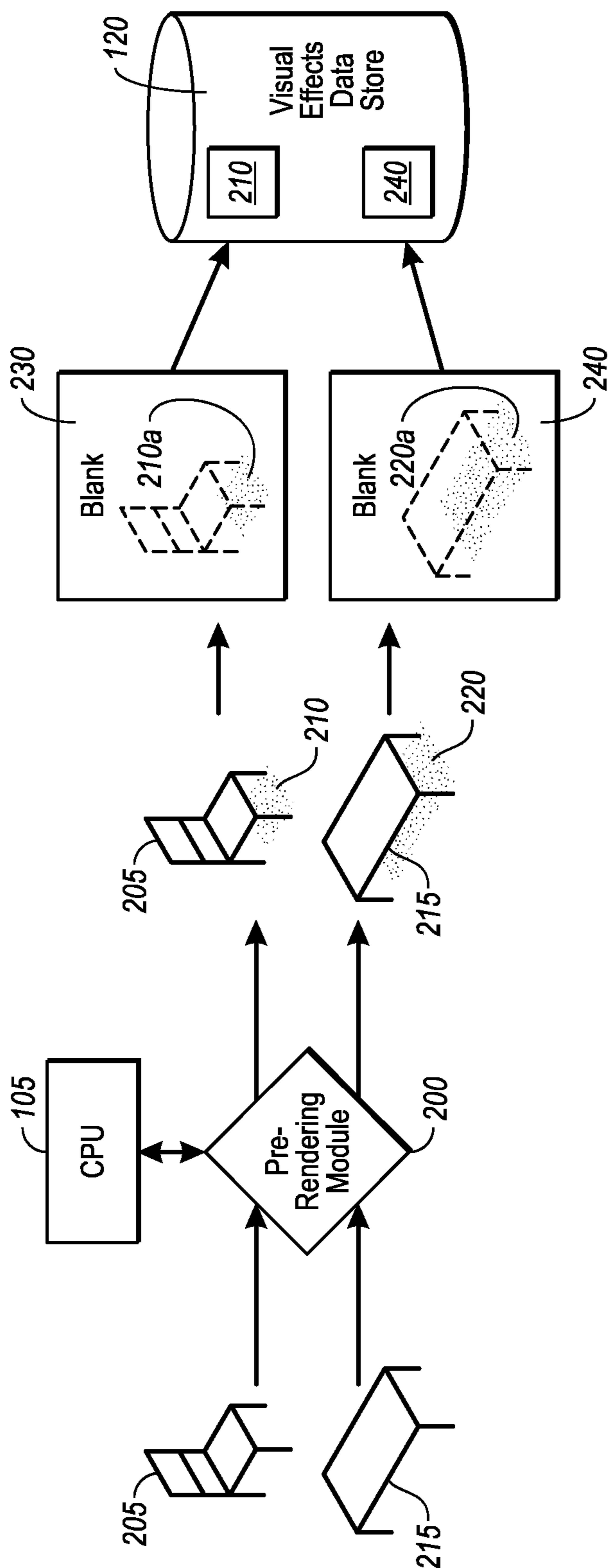


Fig. 2

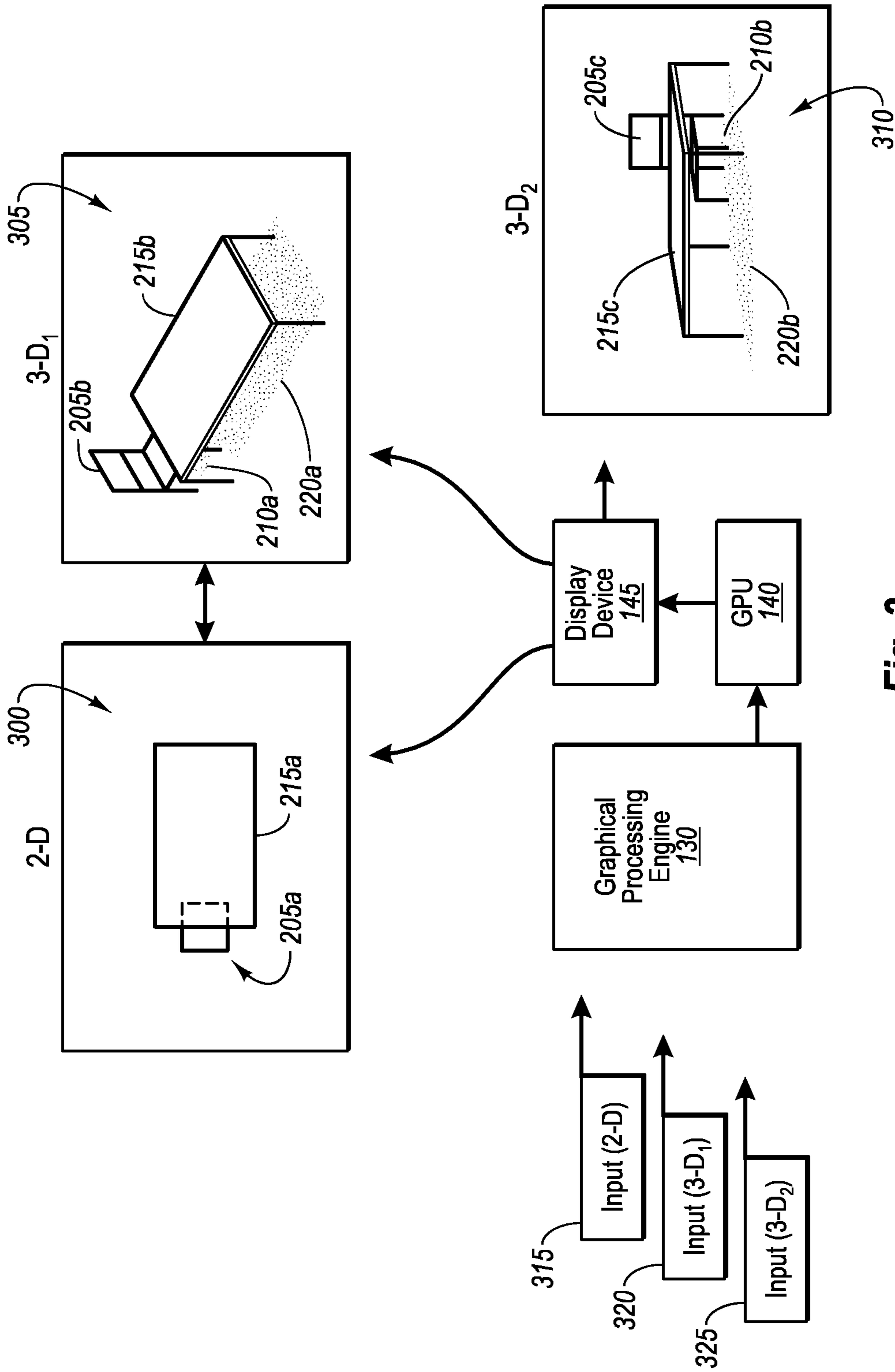
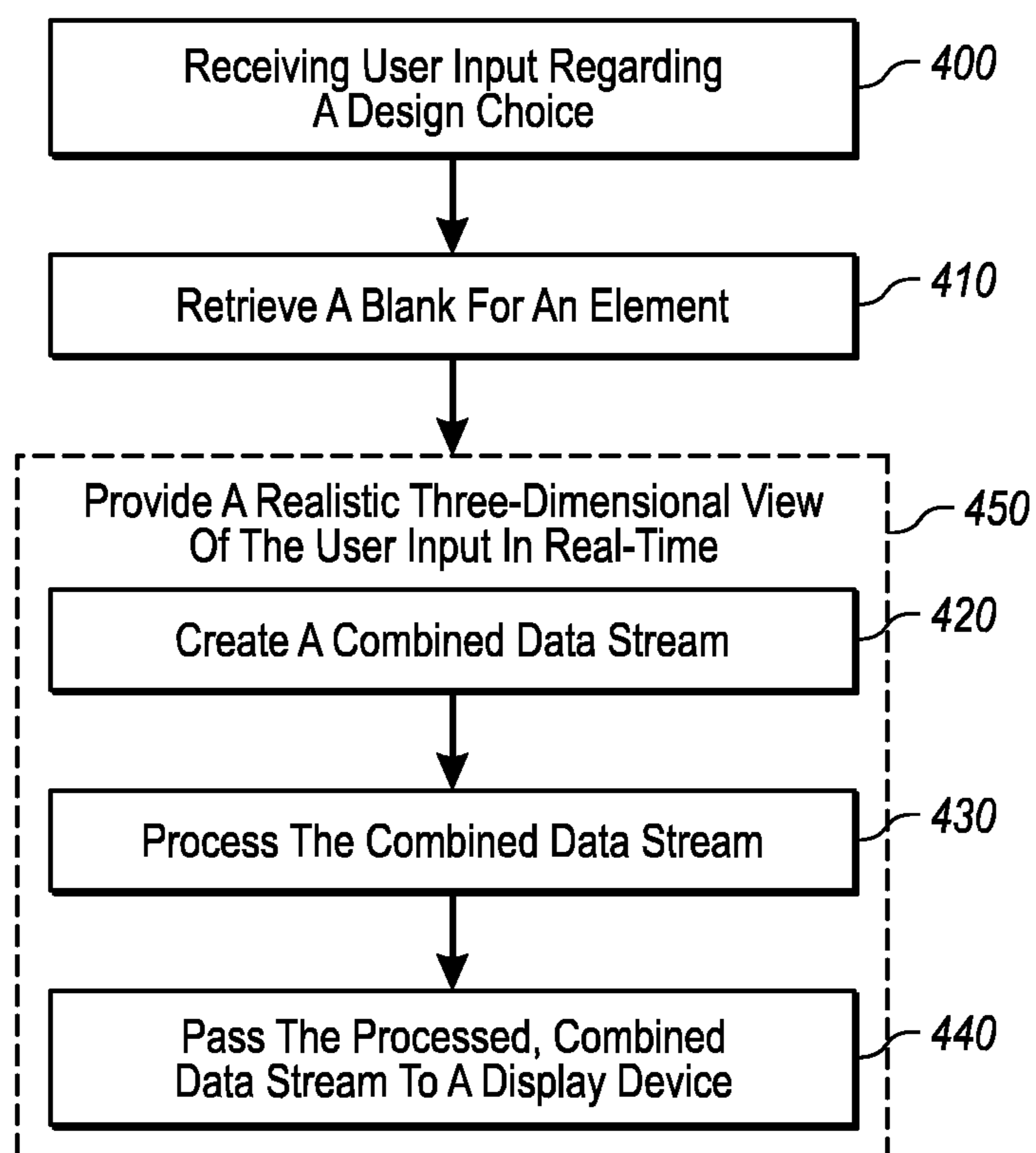
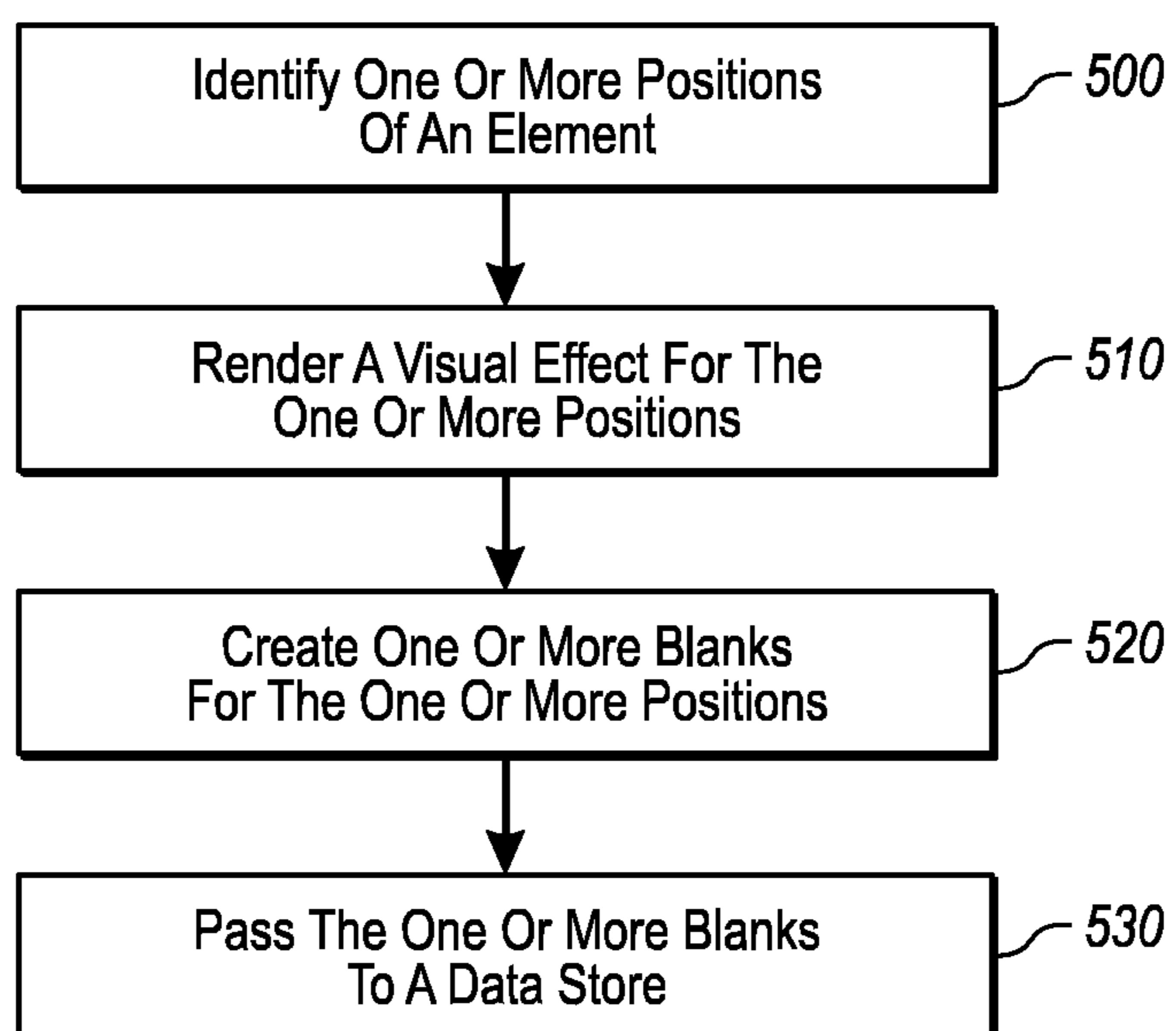


Fig. 3

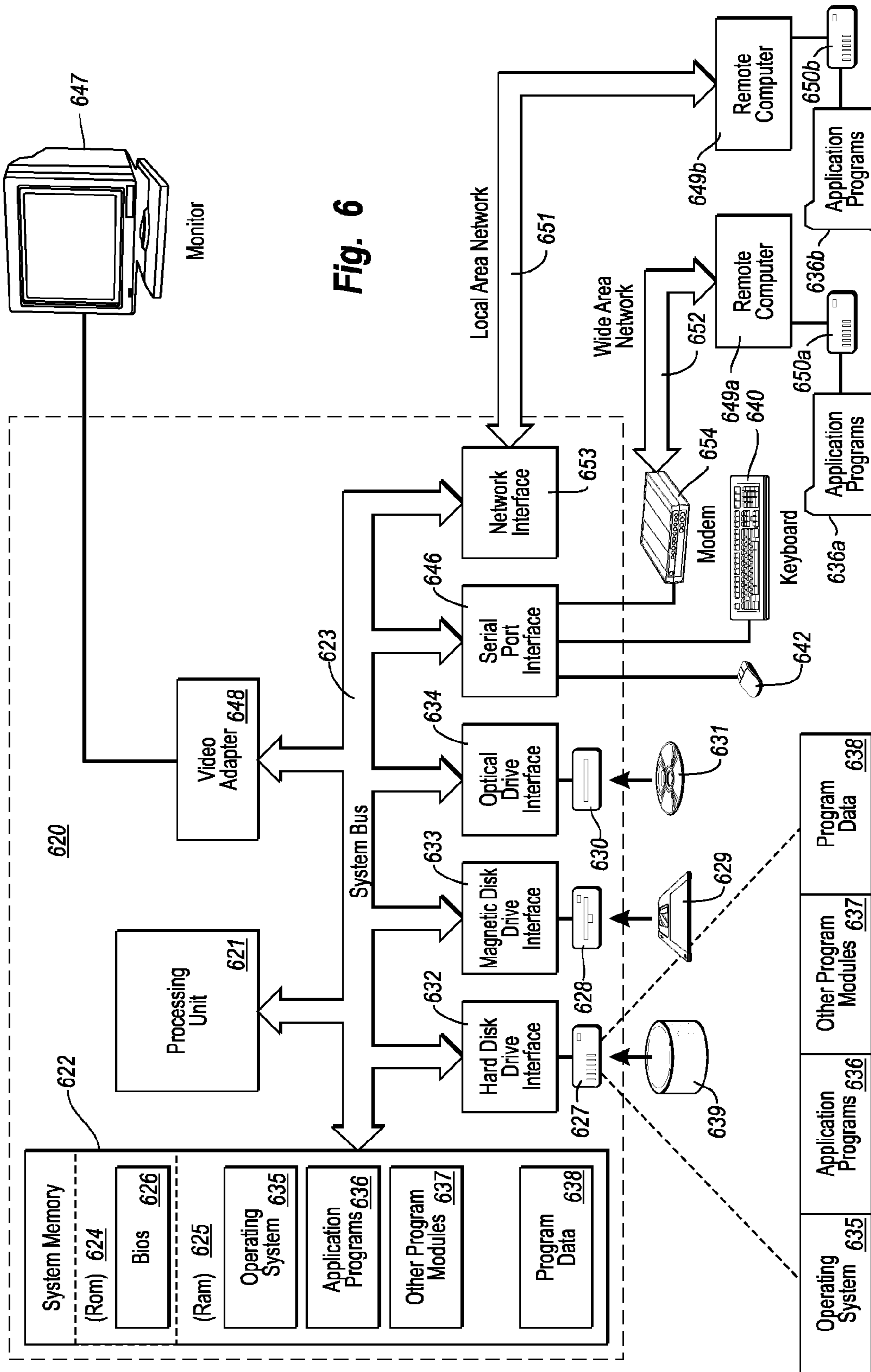


**Fig. 4**





**Fig. 5**



## SOFTWARE INCORPORATING EFFICIENT 3-D RENDERING

### CROSS-REFERENCE TO RELATED APPLICATIONS

The present invention is a continuation of U.S. Non-Provisional patent application Ser. No. 11/204,421, filed on Aug. 16, 2005, entitled "Capturing a User's Intent in Design Software," which Non-Provisional Application claims priority to U.S. Provisional Patent Application No. 60/602,233, filed on Aug. 17, 2004, entitled "Method and Apparatus for the Selection, Organization and Configuration of Products through Object Oriented Design Intent." The entire content of each of the above listed applications are incorporated by reference herein in their entirety.

### BACKGROUND OF THE INVENTION

#### 1. The Field of the Invention

This invention relates to systems, methods, and computer program products for modeling, such as the design of commercial and residential interiors, and related spaces.

#### 2. Background and Relevant Art

As computerized systems have increased in popularity, so has the range of applications that incorporate computational technology. Computational technology now extends across a broad range of applications, including a wide range of productivity and entertainment software. Indeed, computational technology and related software can now be found in a wide range of generic applications that are suited for many environments, as well as fairly industry-specific software.

One such industry that has employed specific types of software and other computational technology increasingly over the past few years is that related to building and/or architectural design. In particular, architects and interior designers (or "designers") use a wide range of design software for designing the aesthetic as well as functional aspects of a given residential or commercial space. In some such cases, the designer might use some software programs that might be better suited for exterior design, and then use other software programs that might be better suited for interior design. For example, a designer might implement one software program to design an overall look of a building, and then use the software to design or position each of the functional components of the building, such as weight-bearing walls, trusses in a roof, positioning of electrical outlets, and so on. The designer might then use another software program, whether separately, or as an add-on to the first software program, to design functional walls for offices, design where to place work stations, design the position of desks, chairs, lamps, and so forth.

When designing the exterior and/or interior of a given residential or commercial space, the designer will ordinarily need to take care that each of the elements in the design are structurally sound when built. This is because typical design software allows spaces to be fairly configurable to suit the user's tastes without specific regard in many cases to whether the design will actually work. For example, one typical software design program might allow an architect to design a roof or ceiling that is ill-suited for the number or type of weight-bearing walls the architect has presently drawn. If the roof were actually constructed as designed by the architect, the roof or ceiling might collapse. In a situation such as this, however, the builder might indicate to the

architect that the design is physically impossible or impractical, and ask for a redesign. This, of course, can lead to any number of inefficiencies.

Part of the problem with many design software programs that can lead to designing physically impractical structures is the notion that many such design problems require some drawing of a space in flat, two-dimensional space. For example, the outside of a building is designed in a view that emphasizes primarily only height and width, while a top ("plan") view of a building is designed in a view that emphasizes primarily only length and width. With views such as these, the designer will either need to independently visualize the three-dimensional spacing, or will need to perform a separate rendering of the design, if the software allows for it.

While three-dimensional rendering is available in some design software, three-dimensional rendering is fairly processing or resource intensive, and can take an additional amount of time. In particular, traditional rendering programs can take anywhere from several minutes to several hours to appropriately render all of the lighting and shadowing characteristics of a given space with any accuracy. Alternatively, another type of rendering program might simply generate only a very rough set of lighting and shadowing characteristics of a given space based primarily on certain assumptions about a given object's shape.

For example, a gaming engine, which is not typically used in design systems, might rely on a graphical processing unit to determine and generate certain rough visual effects in real-time. With this type of system, however, both the determination and rendering are done as the user is making selections in real-time, and, as such, is quite limited in its ability to provide believable, realistic visual effects that would be useful in a design environment. Thus, conventional software is either too processing intensive, or insufficiently processing intensive to efficiently render expected, believable visual effects of design choices in a given space.

In addition, neither the three-dimensional rendering nor the two-dimensional drawing views are designed to accommodate necessary modifications to the objects or walls, based on real-world materials, or other important constraints. For example, a designer might place several L-shaped desks in a work space that are to be arranged back to back against a cubicle wall. In an ordinary environment, positioning the L-shaped desks together might involve a next step of removing a leg where one leg might be shared, or removing a bracket from one of the L-shaped desks for similar reasons. Accordingly, both the two-dimensional views and three-dimensional renderings of conventional design software tends to capture only what is written, and requires the designer to add or remove parts in a specific drawing to reflect real-world usage. This further encumbers the processing, or potential processing, of realistic visual effects for display, particularly in real-time.

Accordingly, an advantage in the art can be realized with systems, methods, and computer program products that provide a user with the ability to efficiently view and navigate realistic-appearing designs in a highly configurable, and yet user-friendly manner. In particular, an advantage can be realized with expert systems that are configured to specifically capture possible or practical configurations of a designer's intent.

### BRIEF SUMMARY OF THE INVENTION

The present invention solves one or more of the foregoing problems in the prior art with systems, methods, and com-

puter program products configured to efficiently render the visual effects for a user's design choice in a two or three-dimensional view in real-time. In particular, implementations of the present invention relate in part to prerendering lighting, shading, shadowing, or other such visual effects through a conventional central processing unit, and then later processing these effects, along with any other relevant information about the user's design choice, at a graphical processing unit during run-time.

For example, a method in accordance with an implementation of the present invention of accurately and efficiently rendering three-dimensional views of a user's design choices involves receiving user input regarding the positioning of a design element in a design space. Generally, the user input includes one or more attributes associated with the design element. For example, the user input can relate to where a desk goes in relation to a wall, as well as preferences for coloring or material, and the like.

The method also involves retrieving a blank for the design element from a data store. The blank will generally be a template of a visual effect for the design element, such as for shading, shadowing, or other visual effects that might be expected for a given position of a design element. This method further involves a step for providing an accurate three-dimensional view of the user input at a display device through communication with a graphical processing unit. This step generally involves the graphical processing unit providing a display device with the appropriate information so that the display device can display accurate visual effect data for the design element.

Another method in accordance with an implementation of the present invention involves prerendering one or more design elements in part by identifying one or more positions of a design element to be placed in a design space. For example, a user or software engineer determines one or more possible positions of a table or chair, and also determines expected visual effects, such as shadowing, for the table or chair in one or more positions. The method also involves rendering a visual effect for each of the one or more positions, and creating one or more blanks corresponding to each of the one or more positions. Generally, the one or more blanks contain data about a corresponding visual effect for the design element, where the visual effect data is separated from other data such as the size, color, or material used for the design element. In addition, the method involves passing the created one or more blanks to a data store. As such, the one or more blanks can later be accessed by a graphical processing unit, in response to user input for the design element.

Accordingly, implementations of the present invention include front-end and run-time (or both) perspectives that ultimately provide a user of design software with a believable, realistic depiction of design choices in real-time, or as the user is designing a design space. This accurate and real-time creation of the user's design choices can ensure elements are placed in physically appropriate locations, and can also ensure that elements are positioned with functional and ergonomic considerations in mind.

Additional features and advantages of exemplary implementations of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by the practice of such exemplary implementations. The features and advantages of such implementations may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features will become more fully apparent from the following descrip-

tion and appended claims, or may be learned by the practice of such exemplary implementations as set forth hereinafter.

#### BRIEF DESCRIPTION OF THE DRAWINGS

In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

FIG. 1 illustrates a schematic diagram in accordance with an implementation of the present invention in which visual effects are passed from a database to a graphical processing unit before being displayed;

FIG. 2 illustrates a schematic diagram in accordance with an implementation of the present invention in which a chair and a table are prerendered into blanks that passed to a data store;

FIG. 3 illustrates a schematic diagram in accordance with an implementation of the present invention in which multiple user inputs are rendered in real-time for two or three-dimensional views;

FIG. 4 illustrates a flow chart of one or more acts of and steps for accomplishing a method of accurately and efficiently rendering three-dimensional views of a user's design choices during run-time;

FIG. 5 illustrates a flow chart of a sequence of acts of a method of prerendering one or more visual effects for one or more selectable elements; and

FIG. 6 illustrates a schematic diagram of a suitable computing environment for practicing one or more implementations of the present invention.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention extends to systems, methods, and computer program products configured to efficiently render the visual effects for a user's design choice in a two or three-dimensional view in real-time. In particular, implementations of the present invention relate in part to prerendering lighting, shading, shadowing, or other such visual effects through a conventional central processing unit, and then later processing these effects, along with any other relevant information about the user's design choice, at a graphical processing unit during run-time.

For example, as will be understood in greater detail in the following description and claims, at least one aspect of the invention relates to front-loading the processing of much of the rendering (i.e., "prerendering") of certain visual effects, which, in some cases can be fairly time-intensive. In particular, rendering of certain types of visual effects can be done by a central processing unit ("CPU") at a computer, where the prerendering can result in one or more templates, or "blanks", which can be later accessed by a graphical processing engine and graphical processing unit. These templates can be stored with the design software, and then retrieved as needed during run-time.

As such, another aspect of the invention relates to passing previously prepared templates, or blanks, to the graphical processing unit ("GPU") of a computerized system, along

with any other relevant information about a user's design choices. In general, GPU and related hardware is often more able to handle the demands that may be needed for some cases of accurate graphical rendering. These and other aspects of the invention are described with greater detail in the following text.

For example, FIG. 1 illustrates an overall schematic diagram of a system in which user design choices are rendered in real-time for a display device. As shown, design software 100 is loaded in memory 110 of a computerized system. The design software 100 includes a reference library 115 component and a visual effects component 120 (or "Blank Data Store").

Generally, the reference library 115 includes all of the physical factors of a given element (such as a design element in an architectural application, or an element of an apparatus). The physical factors stored in the reference library are ultimately correlated with attributes of user input, such as the position, or types of materials that can be used with one type of table or chair versus another, as well as the types of materials that are shared between tables or chairs when put in position. The reference library 115 also includes information regarding possible positioning of elements in a space, such that the design software can prohibit certain sizes of a table, or prohibit a table from being placed on top of a wall, or the like. Information from the reference library 115 component is also combined with user input in the object data 122 component. The object data 122 component includes program objects and related data that are created in conjunction with reference library 115 information based on a sequence of one or more user design choices, and one or more attributes associated with those design choices.

Additional and detailed descriptions of using data objects to capture a user's design intent, as well as more detail of the relationship between user input and attributes, are found in commonly-assigned U.S. patent application Ser. No. 11/204,421, filed on Aug. 16, 2005, entitled "Capturing a User's Intent in Design Software"; U.S. patent application Ser. No. 11/204,419, also filed on Aug. 16, 2005, entitled "Design Software Incorporating Efficient 3-D Rendering"; and in U.S. patent application Ser. No. 11/204,420, also filed on Aug. 16, 2005, entitled "Capturing a User's Design Intent With Resolvable Objects". The entire content of each of the aforementioned patent applications is incorporated by reference herein.

In general, the visual effects data store 120 includes information regarding—as the name implies—possible visual effects for any position of a design element relative to another element such as a floor, table, wall, etc. These visual effects can include shading, shadowing, or other general lighting characteristics for a chosen position of the element. For example, the visual effects data store 120 includes lighting templates for an L-shaped table when put against a wall, as well as a separate visual effects template for the table when free standing in the middle of a room, near a window, or the like. The visual effects data store 120 can also include information regarding orientations of the design element, such as when the table is facing downward, upward, leftward, or rightward, and so forth. FIG. 1 shows that this information, as well as information from the reference library 115 and data objects 120 component are processed by the CPU 105.

This processing, however, contrasts with processing that occurs with respect to real-time rendering of visual effects to a display device for the user's view. For example, FIG. 1 also shows that the design software 100 passes data from the reference library 115, the visual effects data store 120, and

any other related object data 122 to graphical processing engine 130, which may or may not be loaded into the main memory 110. That is, the graphical processing engine 130 may be installed in other implementations on, for example, the graphics hardware on which the GPU 140 is found. Thus, FIG. 1 illustrates an example where the design software 100 and peripheral input device (not shown) processing is handled by central processing unit 105, while processing for visual effect data is handled by graphical processing unit ("GPU") 140.

In particular, FIG. 1 shows that the design software 100 passes data from the reference library 115, data store 120, and the object data 122 to graphical processing engine 130, in response to user input 125 for a specific view. For example, the user might have positioned a certain chair and/or table in a two-dimensional design space (e.g., FIG. 3), and sends an input 125, with one or more corresponding attributes (e.g., position, size, color, material, etc.) to the design software 100 signaling a desire to see the elements in a three-dimensional view. Based on the type of elements (e.g., chair and table) selected, and based on the position of the elements in the design space, the design software 100 determines that the three-dimensional view will include certain visual effects for added realism. Each individual visual effect, in turn, is found in the data store 120.

The design software also identifies, for example, that the user has selected a blue table top (not shown), and so pulls from the reference library 115 the available color and material that either match, or closely match, the user's design choices. The design software further identifies other variables associated with the user's design choice, such as the size (e.g., stretching) or X/Y positioning of the given design element in the design space. The design software 100 then pulls this data from an object data module 122, and then passes it with the data of the reference library 115 and the data store 120 to the graphical processing engine 130.

In general, the graphical processing engine 130 comprises one or more sets of computer-executable code that are configured to prepare data for processing at the GPU 140, such that the GPU processes (i.e., generates) corresponding pixel information to be sent to display 145. In one implementation, the graphical processing engine 130 is similar in some respects to a game engine, which takes data from one program component and passes the data to another program component, as necessary, to identify appropriate pixel information. As shown in FIG. 1, for example, the graphical processing engine 130 receives data from the design software 100, and holds the data passed from the design software in a "scene graph" 135. A scene graph is effectively a data store that coordinates location and other relevant information for each element of data to be rendered.

The graphical processing engine 130 then prepares a combined data stream, and passes along the stream to the GPU 140. GPU 140 then processes the combined data stream separately from other processing components in the computer system, and sends the processed data (e.g., pixel information) to a display device 145. Since much of the detailed rendering of the visual effects has been done previously and turned into a template (or "blank"), the GPU 140 can produce a fairly accurate data stream from the template without requiring a significant amount of additional processing resources and processing time.

At least one result of this process is that the user can view a detailed and realistic-appearing view of the current design choices in the design space almost instantly after requesting it from the design software 100. Furthermore, the user can navigate different visual effects of different angles for the

design elements throughout the three-dimensional view without needing to wait for additional processing. Thus, the design software **100** can provide a much richer viewing experience to the user regarding how various design choices will look in a real-world environment, in much quicker time.

FIG. 2 illustrates a conceptual diagram of an exemplary prerendering phase in accordance with one or more implementations of the present invention. In particular, FIG. 2 shows where a chair **205** and a table **215** are prerendered into one or more visual effects that are stored for subsequent processing in data store **120**. As shown, one position of a chair **205** and one position of a table **215** are passed to prerendering module **200**. These specific positions of the chair **205**, table **215**, or other elements such as a wall (not shown), lamp (not shown), or other such design element will typically be determined in advance by an engineer for the design software **100**. (There may, however, be other instances where the user may want to decide additional visual effects to include with the design software **100**.) For example, the user or engineer might decide that when the chair **205** is placed against a wall (not shown) on one side, the chair will need to have one type of shadow, while the chair will have another shadow when placed against the wall on the opposing side, perhaps since there is an additive or cancelled effect of chair and wall shadows.

The user or engineer might also determine that the chair or table will have still another shadow or visual effect when placed by itself in the middle of a design area. In other cases, the user or engineer simply assumes a consistent, multi-directional light source that causes a consistent shadow for the design element in virtually any position. One will appreciate, therefore, that the possibilities for rendering of various types of visual effects, and for creating corresponding blanks, are essentially endless.

The user or engineer then passes the relevant element positions into a prerendering module **200**. The prerendering module **200** then creates a separate visual effect for each orientation of each element. In particular, different shading or lighting effects for chair **205** in 5 different positions will mean that the prerendering module may render at least 5 separate visual effects that can be translated into separate corresponding blanks (e.g., **230**). In other cases, the prerendering module **200** can generate more or fewer blanks (i.e., reusable for different positions), as needed. The prerendering module **200** in turn uses the CPU **105** for this processing, which can take anywhere from about a few minutes to a few hours for each visual effect.

As shown in FIG. 2, for example, the prerendering module **200** has created (i.e., “rendered”) one or more visual effects (e.g. one or more shadows **210**) for chair **205** and one or more visual effects (e.g., one or more shadows **220**) for table **215**, each shadow being rendered for corresponding orientations or positions of the chair or table. In particular, each shadow **210** or **220** can be a single shadow, or a composite of one or more shadows created from different light sources. In other implementations, the user or design may have also independently rendered multiple positions of separate components of the chair or table for added accuracy. For example, the user or engineer might have independently rendered each chair leg and chair back independently from the stool portion of the chair. Thus, it will be appreciated that the illustrated example shows only a basic shadow visual effect for purposes of convenience.

After rendering the visual effect, the prerendering module **200** creates a “blank” **230** for the chair and a “blank” **240** for the table for the given orientation. In general, a “blank” is an accessible data file that includes a template for the given

visual effect. That is, the blank represents a dissociation of the image of the element (e.g., the image of chair **205**) from the rendered visual effect (e.g., the image of shadow **210**). For example, FIG. 2 shows that blank **230** includes one orientation of a shadow **210a**, and blank **240** includes one orientation of a shadow **220a**. The prerendering module **200** then passes the created blanks **230** and **240** into the visual effects data store **120**, where they can be accessed as needed.

For example, FIG. 3 shows that a first user input **315** includes positioning table **215a** and chair **205a** in a particular position of a two-dimensional view **300**. To implement this view, the graphical processing engine **130** combines the user input as received from the design software, in addition to any other elements that might be important for the two-dimensional view, as described earlier. The graphical processing engine **130** then passes the relevant data to GPU **140** for processing, and the GPU **140** passes a processed data stream to the display device **145**. Thus, input **315** results in “2-D” view **300**.

The user can then view the table **215a** and chair **205a** and move, reposition, or change the design elements however the user sees fit. For example, the user can even change the chair **205a** to another element, such as another table, or can make a selection for another color or material used by the chair or table. Each user change such as this can involve the design software **100** extracting additional information from the object data **122** (or reference library **115**, when appropriate) and passing this information to the graphical processing engine **130**.

When the user selects a three-dimensional view, such as with input **320**, the design software **100** passes any corresponding blanks (e.g., **230** and **240**) to the graphical processing engine **130**, as appropriate. (In other cases, the blanks **230** and **240** were already passed to the graphical processing engine **130**, and the design software **100** simply tells the graphical processing engine **130** to use what it already has been given.) The graphical processing engine **130** then passes the relevant data to the GPU **140**, and the GPU **140** processes the corresponding data stream to the display device **145**. As shown, input **320** therefore results in “3-D<sub>1</sub>” view **305**, which is a three-dimensional view of the chair (i.e., **205b**), and the table (i.e., **215b**) that includes visual effects.

FIG. 3 also shows that the user can navigate through other perspectives of the three-dimensional views, such as view “3-D<sub>2</sub>” **310**. That is, the GPU **140**, in conjunction with the graphical processing engine **130**, allows the user to navigate under tables, around corners, through ceilings, etc., while still effectively providing the expected visual effects. For example, the user provides additional input **325**, which changes X/Y/Z viewing information for the design space. This input **325** can cause the graphical processing engine **130** to provide additional data to the GPU **140** for processing, or can simply tell the GPU **140** to pull other previously-processed data from cache. This additional user input can further cause the graphical processing engine **130** to receive still other blanks from the blank data store **120**. These blanks (not shown) are then processed at GPU **140**, in conjunction with the new user input **325**, as well as previously processed data (e.g., table color or material, etc.).

As with the two-dimensional view **300**, the user can also change the material, color, or other information of the table and chair (or the like) while peering through a specific 3-D view (e.g., **305**, or **310**). In such a case, the graphical processing engine is not likely to need additional blanks for the change in material, but may pull additional information related to material shape or color from the visual effect data

store **120** and/or reference library **115**. Thus, little additional data needs to be processed, resulting in a substantially immediate representation of the new input through the corresponding interface.

Accordingly, the schema shown and described in FIGS. **1-3** illustrate a number of program components, modules, and/or corresponding functions for representing user design choices in a believable, realistic view in an efficient manner.

FIGS. **4** through **5** illustrates non-functional acts and/or functional steps that include non-functional acts for accomplishing one or more methods in accordance with the present invention. In particular, FIG. **4** illustrates a flow chart of one or more acts of and steps for accurately and efficiently rendering three-dimensional views of a user's design choices during run-time. By contrast, FIG. **5** illustrates a flow chart of one or more acts of a method for prerendering one or more visual effects for one or more selectable elements, such that the user's design choices can be rendered in real-time for a realistic display. The methods illustrated in FIGS. **4** and **5** are described below with reference to the preceding FIGS. **1-3**.

For example, FIG. **4** shows that a method of efficiently rendering believable three-dimensional views comprises an act **400** of receiving user input regarding a design choice. Act **400** includes receiving user input regarding the positioning of a design element in a design space, the user input including one or more attributes associated with the design element. For example, a user uses an input device to provide input **125** to design software **100**, where the user input relates to the selection and placement of a chair **205a** and a table **215a** in a two-dimensional design space **300**.

In addition, the method of FIG. **4** comprises an act **410** of retrieving a blank for a design element. Act **410** includes retrieving a blank for the design element from a database. For example, upon receipt of a user input **320** for viewing a three-dimensional view **305** of a design space, the design software **100** and graphical processing engine **130** communicate to exchange one or more blanks (e.g., **230** and **240**), if they have not already been communicated. The one or more blanks are then processed with any other relevant information for the design element for a selected three-dimensional view.

The method illustrated in FIG. **4** also comprises a step for providing a realistic three-dimensional view of the user input in real-time. Step **450** includes providing a realistic three-dimensional view of the user input at a display device through communication with a graphical processing unit, such that the graphical processing unit processes and provides to the display device accurate visual effect data for the design element. For example, when a user selects a three-dimensional view, the user is presented with a seemingly instantly-rendered three-dimensional view of a given design space that has believable-looking visual effects. The user can then navigate throughout various corners and angles of the design space in real-time, without necessarily requiring significant waiting periods for additional processing.

Although step **450** can be accomplished by any number or order of corresponding non-functional acts, FIG. **4** shows that step **450** comprises at least an act **420** of creating a combined data stream. Act **420** includes creating a combined data stream that includes one or more of the blank for the design element and any of the one or more attributes. For example, as shown in FIG. **1**, graphical processing engine **130** receives data from the design software **100** relating to object data **122**, any other information from a reference library **115**, as well as any visual effects information from a blank data store **120**. In addition, step **450** comprises an act

**430** of processing the combined data stream. Act **430** includes processing the combined data stream at the graphical processing unit. For example, the graphical processing engine **130** passes the combined data stream to GPU **140**, where the data is processed separately from the hardware that is processing the design software **100** in the computer system.

Furthermore, FIG. **4** shows that step **450** comprises an act **440** of passing the processed combined data stream to a display device. Act **440** includes passing the processed combined data stream to the display device upon selection of a three-dimensional view. For example, as shown in FIG. **3**, after receiving input **320**, the GPU **140** processes and passes generated pixel information to the display device **145**, such that the display device **145** can show three-dimensional view **305**. Accordingly, the method of FIG. **4** provides a user with the ability to design an interior or exterior space and also efficiently view that space in any of two or three dimensional views in an accurate manner without undue delay.

FIG. **5** illustrates another method in accordance with an implementation of the present invention, albeit from the perspective of prerendering design elements before they are processed by the GPU **140**. In particular, FIG. **5** shows that a method of prerendering one or more visual effects for one or more selectable design elements, such that the user's design choices can be accurately rendered in real-time comprises an act **500** of identifying one or more positions of a design element. Act **500** includes identifying one or more positions of a design element to be placed in a design space. For example, as shown in FIG. **2**, a user or software engineer will identify one or more positions, angles, or the like for a design element (e.g., chair **205**, table, **215**) as it is placed in a design space, such as how it is positioned next to a wall, door, or on an assumed floor, and determine a corresponding visual effect. The user or engineer might determine that a realistic shadow will face in one direction and in a certain shape when the design element is placed beside a wall on one side, and will face in another direction and in another shape when the design element is placed on the other side of the wall.

The method of FIG. **5** also comprises an act **510** of rendering a visual effect for the one or more positions. Act **510** includes rendering a visual effect for each of the one or more positions. For example, the user or engineer passes the information about desired lighting or other visual effects for each design element (e.g., chair **205**, table **215**) into a prerendering module **200**. Typically, a CPU **105** spends as much as a few minutes to one or more hours rendering each selected visual effect for each of the one or more positions of the selected design elements.

In addition, the method of FIG. **5** comprises an act **520** of creating one or more blanks for the one or more positions. Act **520** includes creating one or more blanks corresponding to each of the one or more positions, the one or more blanks containing data about a corresponding visual effect for the design element. For example, the prerendering module **200** creates a visual effect of a shadow **210** for one position of chair **205** and a shadow **220** for one position of table **215**. The prerendering module then prepares corresponding one or more blanks (e.g., **230**) for the chair shadow **210a** and one or more blanks (e.g., **240**) for the table shadow **220a**. The one or more blanks (e.g., **230**, **240**) essentially separate the image of the design element (e.g., chair **205** or table **215**) from the visual effect, such that the file is primarily of an orientation of the visual effect by itself.

The method of FIG. **5** also comprises an act **530** of passing the one or more blanks to a data store. Act **530**

includes passing the created one or more blanks to a data store, such that the one or more blanks can later be accessed by a graphical processing unit, in response to user input for the design element. For example, the prerendering module **200** passes blanks **230, 240** to data store **120**. Thus, at a later point, and if the graphical processing unit **130** has not already received the corresponding blanks, the graphical processing unit **130** accesses the blanks **230, 240** via the design software **100**, or directly from the data store **120** in response to user input **320** for a three-dimensional view. Thus, most, if not all, of the processing-intensive visual effects for the design elements accessed during run-time are prerendered, and made accessible through a data store. Accordingly, the method of FIG. **5** illustrates at least one way in which data that would otherwise be difficult or impossible to process in real-time can be made available, and thereby provide the user of design software with essentially instantaneous, realistic-looking views of design choices.

The foregoing schema and methods, therefore, provide designers and users with a wide variety of options for designing and viewing interior spaces. In particular, implementations of the present invention allow design spaces to be prepared and viewed quickly in a believable way, so that the designer can efficiently view how design choices will look in an actual setting. Furthermore, the provisions for essentially instant rendering of certain design effects allow the user to make better informed decisions about what design elements should go in certain places and/or in certain orientations in real-time.

As previously described, this real-time aspect can be accomplished by separately rendering complex visual effects at a CPU, and then later rendering these effects with other information at run-time in a GPU. It will be appreciated, however, that, although much of the foregoing discussion has focused on separate processing by different hardware components (e.g., CPU and GPU), separate rendering is not necessarily required. In particular, a computer system could be configured with sufficient processing power to prepare detailed, accurate visual effects, and combine those visual effects with user input pursuant to sending the data to output. For example, the design software **100** can process the intended visual effect in the CPU **105**, hand off the processed visual effect to a GPU **140**, and then pass the requested output in what would appear to the user as a relatively instantaneous amount of time. Thus, separate processing is only one way, albeit one convenient way, of accomplishing one or more ends of the invention.

In addition, although the discussion herein has related primarily to architectural-related design choices, implementations of the present invention are not necessarily limited thereby. In particular, the design software of the present invention can readily be configured for a wide variety of uses, such as for designing tools, machines, or other types of systems where visual effect information could be useful for positioning certain elements relative to other elements in a given space or inside an apparatus. Accordingly, it will be appreciated that the general principles articulated herein of rendering and prerendering in various stages for creating believable visual effects in real-time can have potentially wide application.

FIG. **6** and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by computers in network

environments. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Computer-executable instructions, associated data structures, and program modules represent examples of the program code means for executing steps of the methods disclosed herein. The particular sequence of such executable instructions or associated data structures represents examples of corresponding acts for implementing the functions described in such steps.

Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including personal computers, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where local and remote processing devices perform tasks and are linked (either by hardwired links, wireless links, or by a combination of hardwired or wireless links) through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to FIG. **6**, an exemplary system for implementing the invention includes a general-purpose computing device in the form of a conventional computer **620**, including a processing unit **621**, a system memory **622**, and a system bus **623** that couples various system components including the system memory **622** to the processing unit **621**. The system bus **623** may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) **624** and random access memory (RAM) **625**. A basic input/output system (BIOS) **626**, containing the basic routines that help transfer information between elements within the computer **620**, such as during start-up, may be stored in ROM **624**.

The computer **620** may also include a magnetic hard disk drive **627** for reading from and writing to a magnetic hard disk **639**, a magnetic disc drive **628** for reading from or writing to a removable magnetic disk **629**, and an optical disc drive **630** for reading from or writing to removable optical disc **631** such as a CD ROM or other optical media. The magnetic hard disk drive **627**, magnetic disk drive **628**, and optical disc drive **630** are connected to the system bus **623** by a hard disk drive interface **632**, a magnetic disk drive-interface **633**, and an optical drive interface **634**, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer-executable instructions, data structures, program modules and other data for the computer **620**. Although the exemplary environment described herein employs a magnetic hard disk **639**, a removable magnetic disk **629** and a removable optical disc **631**, other types of computer readable media for storing data can be used, including magnetic cassettes, flash memory cards, digital versatile disks, Bernoulli cartridges, RAMs, ROMs, and the like.

Program code means comprising one or more program modules may be stored on the hard disk **639**, magnetic disk **629**, optical disc **631**, ROM **624** or RAM **625**, including an operating system **635**, one or more application programs **636**, other program modules **637**, and program data **638**. A user may enter commands and information into the computer **620** through keyboard **640**, pointing device **642**, or other input devices (not shown), such as a microphone, joy



stick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 621 through a serial port interface 646 coupled to system bus 623. Alternatively, the input devices may be connected by other interfaces, such as a parallel port, a game port or a universal serial bus (USB). A monitor 647 or another display device is also connected to system bus 623 via an interface, such as video adapter 648. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

The computer 620 may operate in a networked environment using logical connections to one or more remote computers, such as remote computers 649a and 649b. Remote computers 649a and 649b may each be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically include many or all of the elements described above relative to the computer 620, although only memory storage devices 650a and 650b and their associated application programs 636a and 636b have been illustrated in FIG. 6. The logical connections depicted in FIG. 6 include a local area network (LAN) 651 and a wide area network (WAN) 652 that are presented here by way of example and not limitation. Such networking environments are commonplace in office-wide or enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 620 is connected to the local network 651 through a network interface or adapter 653. When used in a WAN networking environment, the computer 620 may include a modem 654, a wireless link, or other means for establishing communications over the wide area network 652, such as the Internet. The modem 654, which may be internal or external, is connected to the system bus 623 via the serial port interface 646. In a networked environment, program modules depicted relative to the computer 620, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing communications over wide area network 652 may be used.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.

I claim:

1. A computer-implemented method of controlling an architectural design environment in order to more efficiently render three-dimensional views of a user's design choices, the computer-implemented method being performed by one or more processors executing computer executable instructions for the computer-implemented method, and the computer-implemented method comprising:

for each of a plurality of architectural design elements that are selectable for placement within a design space, storing in a reference library data that defines a plurality of physical factors used to define characteristics of each design element;

for one or more of the architectural design elements, inputting to a prerendering module of a central processing unit (CPU) one or more positions and visual effects associated with the one or more positions, and the prerendering module of the CPU generating for

each of the one or more positions a 3-D view containing visual effects, wherein each said 3-D view is stored in a corresponding template for a given position so that 3-D views containing visual effects are rendered by the CPU and stored for later retrieval by a graphics processing unit (GPU), and wherein each template comprises an accessible data file stored in a visual effects data store;

displaying the design space on a computerized display; selecting an architectural design element and retrieving from the reference library the data that defines physical factors for the selected architectural design element;

at an object data component, receiving from a user one or more user inputs representing design choices regarding the selected architectural design element, the one or more user inputs defining attributes associated with the design choices, and the object data component combining the attributes associated with the design choices with the selected architectural design element;

inputting to a graphical processing engine of the GPU

(i) object data for the selected architectural design element and the attributes combined by the object data component with the data that defines the physical factors used for the selected architectural design element, and

(ii) a template of a 3-D view retrieved from the visual effects store corresponding to a given position for the selected architectural design element;

the graphical processing engine of the GPU processing the object data to prepare a rendering of the object data for the selected architectural design element;

preparing, with the graphical processing engine, a combined data stream comprised of the processed object data for the selected architectural design element and the input template of the 3-D view retrieved from the visual effects store corresponding to the given position for the selected architectural design element, wherein the graphical processing engine of the GPU is only required to process the object data but not the pre-rendered 3-D view retrieved from the visual effects store;

the graphical processing engine of the GPU generating pixel information from the combined data stream; and sending the pixel information to the computerized display for display in the generated design space.

2. The computer-implemented method as recited in claim 1, further comprising:

storing the processed object data within a scene graph, and

wherein the stored processed object data relates to locations and other attributes of one or more architectural design elements.

3. The computer-implemented method as recited in claim 1, wherein the graphical processing engine comprises a game engine.

4. The computer-implemented method as recited in claim 1, wherein at least one or more 3-D views stored in a template comprises a pre-rendered shadow for a three-dimensional piece of furniture.

5. The computer-implemented method as recited in claim 1, wherein the template for each position is defined by a particular position with respect to a light source.

6. The computer-implemented method as recited in claim 5, wherein the 3-D view of a visual effect stored in at least one template is pre-rendered from a plurality of different positions with respect to a light source.

7. A computer program storage product storing computer-executable instructions for a method of controlling an architectural design environment in order to more efficiently render three-dimensional views of a user's design choices, the method being performed by one or more processors executing the computer executable instructions, and the method comprising:

for each of a plurality of architectural design elements that are selectable for placement within a design space, storing in a reference library data that defines a plurality of physical factors used to define characteristics of each design element;

for one or more of the architectural design elements, inputting to a prerendering module of a central processing unit (CPU) one or more positions and visual effects associated with the one or more positions, and the prerendering module of the CPU generating for each of the one or more positions a 3-D view containing visual effects, wherein each said 3-D view is stored in a corresponding template for a given position so that 3-D views containing visual effects are rendered by the CPU and stored for later retrieval by a graphics processing unit (GPU), and wherein each template comprises an accessible data file stored in a visual effects data store;

displaying the design space on a computerized display; selecting an architectural design element and retrieving from the reference library the data that defines physical factors for the selected architectural design element;

at an object data component, receiving from a user one or more user inputs representing design choices regarding the selected architectural design element, the one or more user inputs defining attributes associated with the design choices, and the object data component combining the attributes associated with the design choices with the selected architectural design element;

inputting to a game engine of the GPU

(i) object data for the selected architectural design element and the attributes combined by the object data component with the data that defines the physical factors used for the selected architectural design element, and

(ii) a template of a 3-D view retrieved from the visual effects store corresponding to a given position for the selected architectural design element;

processing the object data with a game engine, wherein the game engine renders the object data and identifies corresponding pixel information;

preparing, with the game engine, a combined data stream comprised of the processed object data for the selected architectural design element and the input template of the 3-D view retrieved from the visual effects store corresponding to the given position for the selected architectural design element, wherein the game engine of the GPU is only required to process the object data but not the pre-rendered 3-D view retrieved from the visual effects store;

the game engine of the GPU generating pixel information from the combined data stream; and

sending the pixel information to the computerized display for display in the generated design space.

8. The computer program storage product as recited in claim 7, wherein one or more architectural design elements comprise a piece of furniture.

9. The computer program storage product as recited in claim 8, wherein one or more of the pre-rendered templates comprise a three-dimensional effect associated with a piece of furniture.

10. The computer program storage product as recited in claim 8, wherein the 3-D view contained in each stored template of a visual effect is associated with a particular position of a given architectural design element with respect to a light source.

11. The computer program storage product as recited in claim 10, wherein multiple templates of visual effects for a single architectural design element are pre-rendered for a variety of different light configurations.

12. The computer program storage product as recited in claim 8, wherein the user can change the color of an element associated with a given template of a visual effect for a given architectural design element.

13. The computer program storage product as recited in claim 8, wherein the user can change the material of a given architectural design element associated with a template.

14. The computer program storage product as recited in claim 7, wherein the CPU and GPU are provided by a single processor.

15. The computer program storage product as recited in claim 14, wherein the CPU and GPU are provided by different processors.

16. A computer system for an architectural design environment, comprising:

a reference library that stores for each of a plurality of architectural design elements that are selectable for placement within a design space data that defines a plurality of physical factors used to define characteristics of each design element;

a prerendering module of a central processing unit (CPU) that receives for one or more of the architectural design elements one or more positions and visual effects associated with the one or more positions, and the prerendering module of the CPU generating for each of the one or more positions a 3-D view containing visual effects, and wherein each said 3-D view is stored in a corresponding template for a given position so that 3-D views containing visual effects are rendered by the CPU and stored for later retrieval by a graphics processing unit (GPU), and wherein each template comprises an accessible data file stored in a visual effects data store;

a computerized display that displays the design space; an object data component that receives from a user one or more user inputs representing design choices regarding a selected architectural design element, the one or more user inputs defining attributes associated with the design choices, and the object data component combining the attributes associated with the design choices with the selected architectural design element;

a graphical processing engine of the GPU that receives (i) object data for the selected architectural design element and the attributes combined by the object data component with the data that defines the physical factors used for the selected architectural design element, and

(ii) a template of a 3-D view retrieved from the visual effects store corresponding to a given position for the selected architectural design element;

the graphical processing engine of the GPU processing the object data to prepare a rendering of the object data for the selected architectural design element;

wherein the graphical processing engine then prepares  
 a combined data stream comprised of the processed  
 object data for the selected architectural design ele-  
 ment and the input template of the 3-D view  
 retrieved from the visual effects store corresponding 5  
 to the given position for the selected architectural  
 design element, and wherein the graphical process-  
 ing engine of the GPU is only required to process the  
 object data but not the pre-rendered 3-D view  
 retrieved from the visual effects store; 10  
 the graphical processing engine of the GPU generating  
 pixel information from the combined data stream;  
 and  
 the graphical processing engine of the GPU sending the  
 pixel information to the computerized display for 15  
 display in the generated design space.

**17.** The computer system as recited in claim **16**, wherein  
 one or more architectural design elements comprise a piece  
 of furniture.

**18.** The computer system as recited in claim **17**, wherein 20  
 one or more of the pre-rendered templates comprise a  
 three-dimensional effect associated with a piece of furniture.

**19.** The computer system as recited in claim **17**, wherein  
 the 3-D view contained in each stored template of a visual  
 effect is associated with a particular position of a given 25  
 architectural design element with respect to a light source.

**20.** The computer system as recited in claim **19**, wherein  
 multiple templates of visual effects for a single architectural  
 design element are pre-rendered for a variety of different  
 light configurations. 30

\* \* \* \* \*